

# Rendering and Animating Expressive Caricatures

Mohammad Obaid<sup>\*†</sup>, Ramakrishnan Mukundan<sup>†</sup>, and Mark Billinghurst<sup>\*</sup>

<sup>\*</sup>HITLab New Zealand, University of Canterbury, Christchurch, New Zealand

<sup>†</sup>Computer Science and Software Engineering Dept., University of Canterbury, New Zealand

Email: {mohammad.obaid, mark.billinghurst}@hitlabnz.org,

mukund@cosc.canterbury.ac.nz

**Abstract** — We present a novel approach for generating stylized artistic rendering of caricatures from a given face image, with the ability to map any one of the main six expressions and control the degree of its expressiveness on the generated caricature. Our method achieves this by manipulating the facial appearance and expressivity of the caricature, using quadratic deformation model representations of facial expressions. A stroke-based non-photorealistic rendering (NPR) engine is developed to generate the output caricature that appears to be a sketch of the original image. The NPR engine segments the defined rendering path and generates rectangular strokes that approximate each segment. The stroke properties are computed by applying geometric image moments on each segment. To enhance the overall appearance of the caricature, extra strokes are randomly positioned around the rendering path. The results show that from an input neutral facial image, expressive rendered caricatures are generated for the facial expressions neutral, smile, sad, fear, surprise, disgust, and anger. Our approach produces very expressive artistically rendered caricatures, and could lead to future research directions in further extending the method to produce pencil sketch rendering of images or rendering 3D characters to generate cartoon-like 3D characters.

**Keywords**—Expressive Caricatures, Non-photorealistic Rendering, Caricature Sketch Rendering, Facial Caricature

## I. INTRODUCTION

Facial caricature drawing exaggerates physical face features for a comical effect, and creates an entertaining, humorous, and cartoon-like description of a person's face. Many skilled artists have contributed to the field of caricature illustrations and drawings, such as the work by Hughes and Gair [1].

Recently, computerized caricatures have been introduced as part of computer graphics' non-photorealistic rendering technologies. Research has been conducted on how to produce caricatures automatically or semi-automatically using various computer graphics techniques. Most of these methods train their systems on how to generate and exaggerate stylized caricatures by using real artist prototype sketch drawings. One of the limitations to these methods is that they require a lot of manual work with a large number of training prototypes drawn by artists. In addition, the final appearance of the caricature is limited to the prototypes used in the training phase.

In this paper we propose an approach to generate and render facial caricatures from an input face image. Our approach allows the user to control the exaggeration level of the facial expression imposed on the caricature drawing. It is also capable of producing an expressive facial animation of the caricature drawing as shown in Fig. 11. The three

main contributions of our approach are: (1) A quadratic deformation model for the transformation of feature points, which can effectively map any of the six expressions to a face image, (2) an interpolation method for manipulating the facial appearance and expressivity of the caricature, and (3) an automatic moment-based stroke rendering algorithm to render extracted facial features.

The rest of the paper is organized as follows. Section II gives a review of related work and outlines limitations of existing methods. Section III illustrates an overview of the main system components. Section IV explains the steps to extract the rendering paths. In section V, the expressivity control and the exaggeration techniques are described. Section VI gives a description of the stroke-based non-photorealistic rendering algorithm used. Finally, we show the results of our system and conclude the paper in sections VII and VIII respectively.

## II. RELATED WORK

Brenann [2] presented the first interactive caricature generation method that produced exaggerated sketches. More recently, Gooch et al. [3] presented a facial black-and-white illustration system based on the use of Gaussian filters and thresholding techniques. Their system is also capable of generating caricatures by imposing a deformable 2D grid over the surface of the face illustration. Users are then able to interactively change the shape of the whole facial illustration structure. However, most attempts achieve their non-photorealistic look by applying filtering algorithms or building tools for users to imitate traditional artistic styles. There are rare attempts to generate artistic drawings of caricatures using stroke-based rendering algorithms.

Alternatively, Chen et al. [4], presented an example-based facial sketch generation system. Their work analyses and learns the sketch styles of artists and applies them to facial images. Chiang et al. [5] showed a method for generating artistic caricatures used in defining facial feature points based on the MPEG-4 Facial Animation Parameters (FAPs). Most example-based approaches have used statistical analysis and facial feature tracking techniques, such as active appearance models (AAM), to extract the shape of the facial characteristics. Several other approaches have emerged to use statistical models as a tool to extract facial features, such as the work of Su et al. [6], who produced a method to sketch faces and animate them. Their approach first generates a painting

path using an active shape model, and then applies free form deformation to a pen-and-ink line template for the path rendering. Most of these approaches exaggerate the caricature appearance by altering the overall facial shape based on capturing artists' exaggeration prototypes. Rare attempts have been made to alter and control the facial expressions of the generated caricatures. Moreover, most example-based approaches have produced satisfactory results; however, large numbers of reference-caricature sketches drawn by artists are needed to train those systems to create expressive caricatures.

Our system differs from previous work in that it enables the creation and control of artistically rendered expressive facial caricatures based on real facial muscle deformation representations and does not require any earlier training of caricature images drawn by artists.

### III. SYSTEM OVERVIEW

We now briefly describe the main components of our caricature generation system, as shown in Fig. 1. The developed system employs a combination of feature extraction algorithms, facial expression representations using quadratic deformation models, and a stroke-based caricature rendering algorithm.

The aim is to create an expressive caricature from a given neutral face image. The first step is to extract features from the image that will undergo the deformations. AAM's and digital matting are used to identify facial features of interest, such as eyebrows, eyes, nose, lips, ears and hair. We call this part rendering path extraction, as the extracted path will serve as the painting line for the caricature rendering algorithm.

Once the features have been extracted, they are transformed using the facial expression quadratic deformation models [7]. Finally, the caricature rendering engine overlays strokes on the top of the extracted path to produce an amusing caricature that appears to be a sketch of the original image. However in the output image the subject's expressions have been noticeably altered or exaggerated. The following sections explain each of the main parts of our method in more detail.

### IV. RENDERING PATH EXTRACTION

The strokes rendering path represents the location of the features where the strokes will be painted to create the final caricature appearance. The NPR rendering engine, described in section VI, takes as an input the strokes rendering path and generates the final caricature appearance. The strokes rendering path for the hair and the facial features are generated independently. To extract the rendering path of the hair<sup>1</sup>, we use digital matting and edge detection algorithms, while AAM facial feature extraction is used to determined the strokes rendering path for the facial parts. The following sections describe the process in more detail.

<sup>1</sup>The shape of the hair also includes the shape of the ears.

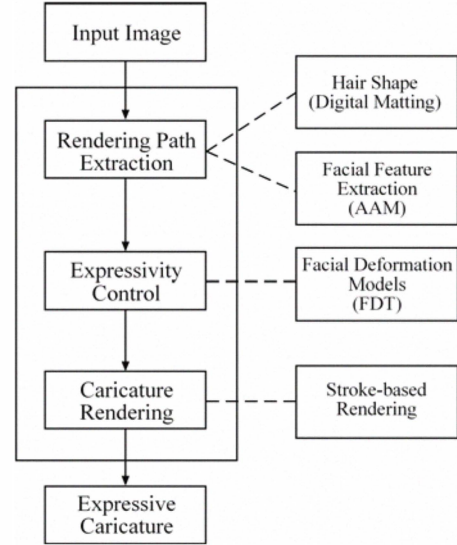


Fig. 1. Caricature system components.

#### A. Hair and ears shape

The shape of the hair can vary from one individual to another, and for that reason it is processed independently from the facial feature parts. To segment the hair from the face and background regions, we use the digital matting algorithm of Sindeyev and Konushin [8]. The method is first initialized by roughly marking out the region of interest (i.e. hair). Digital matting is then used to segment the hair region out. The edges of the hair regions (detected using the canny algorithm [9]) serve as the strokes rendering path. Fig. 2 illustrates the general process.

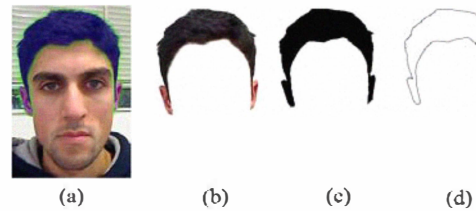


Fig. 2. Hair and Ear shape segmentation. (a) marked region of interest (b) digital matting segmentation (c) threshold image (d) edge detection (strokes rendering path).



Fig. 3. Facial Feature Extraction Using AAM.

## B. Facial Feature Extraction

The strokes rendering paths for the facial features are generated by extracting the facial feature shapes using AAM tracking. This process is carried out in two main steps: a training step and an AAM fitting step. The training step generates a model that can be used for extracting the desired facial features. The facial images for the neutral expression of 30 subjects, acquired from the Cohn-Kanade Database [10], are annotated using the AAM annotation software tools provided by Tim Cootes [11]. For the AAM fitting step we use the algorithm described by Saragih and Goecke [12] to locate and extract the facial feature parts. Fig. 3 demonstrates an example of facial feature extraction using AAM.

## V. EXPRESSIVITY OF THE FACIAL CARICATURE

In this section, we describe how the quadratic deformation model representations of facial expressions can be used to control the expressivity of the caricature drawings. Obaid et al. [7] represented facial expressions by capturing the non-linear nature of muscle deformations for each expression. The following three steps are used in capturing the muscle deformations:

- 1) Subdivide the face into 16 muscle based facial regions, as shown in Fig. 4.
- 2) Use the most general second degree rubber-sheet transformation to derive the deformation parameters for each region by applying the least-square minimization technique.
- 3) Construct Facial Deformation Tables (FDT) to mathematically represent each expression. The procedure to compute a  $FDT_E$  for each of the six main expressions  $E$  (happy, sad, fear, surprise, anger and disgust) is described in [7].

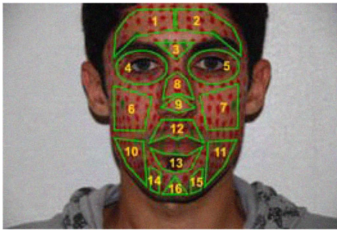


Fig. 4. Facial Muscle Deformation Regions.

In our approach, we map the defined muscle based facial regions into the corresponding extracted facial feature line. This allows us to generate different deformations and exaggerations of the extracted facial features and form the desired caricature appearance. The following sections describe in more detail how the expressivity of the facial features are controlled.

### A. Controlling exaggeration levels

From Obaid et al. [7], rubber-sheet transformations are higher-order (non-linear) polynomial transformations. The name comes from the logical analogy of overlaying an elastic

piece of rubber to fit over a surface of some shape. In the two-dimensional space, rubber-sheet transformations are defined by a pair of equations:

$$x'_i = a_1x_i^2 + a_2x_iy_i + a_3y_i^2 + a_4x_i + a_5y_i + a_6 \quad (1)$$

$$y'_i = b_1x_i^2 + b_2x_iy_i + b_3y_i^2 + b_4x_i + b_5y_i + b_6 \quad (2)$$

$$i = 1, \dots, n$$

where  $n$  is the number of transformed points and  $a_i, b_i$  are the transformation parameters.

Obaid et al. [7] describe how the deformation parameters are derived using a least-square minimization technique, and how it is used to construct a Facial Deformation Table (FDT) to mathematically represent each of the facial expression.

We apply the derived FDT parameters on any feature line of the caricature by first identifying the region to which the line belongs, and then applying the corresponding transformation parameters to every point on that line. Moreover, the way the FDTs are defined allows us to control the expressiveness of a caricature and its level of exaggeration. This is done by interpolating the FDTs parameter values between the natural expression and the desired expression. Extrapolating the parameters beyond the desired expression will generate an exaggeration of the appearance of the facial feature parts. In this context, Equations (5) and (6) can be re-written as follows:

$$x'_{i(E)} = \sum_{u=0}^2 \sum_{v=0}^2 A_{uv(E)} x_i^u y_i^v \quad (3)$$

$$y'_{i(E)} = \sum_{u=0}^2 \sum_{v=0}^2 B_{uv(E)} x_i^u y_i^v \quad (4)$$

where suffix  $(E)$  denotes a transformation specific to an expression  $E$ , and  $A_{00} = a_6$ ,  $A_{01} = a_5$ ,  $A_{10} = a_4$ ,  $A_{02} = a_3$ ,  $A_{11} = a_2$ ,  $A_{20} = a_1$ , with similar mapping for coefficients  $b_i$ .

A parametric linear interpolation between a neutral expression  $N$ , and a given expression  $E$  is then given by

$$\begin{aligned} x'_i &= (1-t)x'_{i(N)} + tx'_{i(E)} \\ &= \sum_{u=0}^2 \sum_{v=0}^2 ((1-t)A_{uv(N)} + tA_{uv(E)}) x_i^u y_i^v \end{aligned} \quad (5)$$

$$\begin{aligned} y'_i &= (1-t)y'_{i(N)} + ty'_{i(E)} \\ &= \sum_{u=0}^2 \sum_{v=0}^2 ((1-t)B_{uv(N)} + tB_{uv(E)}) x_i^u y_i^v \end{aligned} \quad (6)$$

$$0 \leq t \leq 1$$

The parameter  $t$  defines the degree of exaggeration of the expression  $E$ , used for producing the caricature. The value of  $t$  could be increased above 1 to extrapolate the coefficients beyond the normal values for that expression. Our system allows the user to interpolate the parameter  $t$  by varying the value, interactively using a slider, between  $0 \leq t \leq 1$  or extrapolate the value  $t > 1$  (as demonstrated in Fig. 5). A continuous variation of the parameter  $t$  produces an animation sequence for the given facial expression. An example of a

caricature animation for the expression smile is shown in Fig. 11.

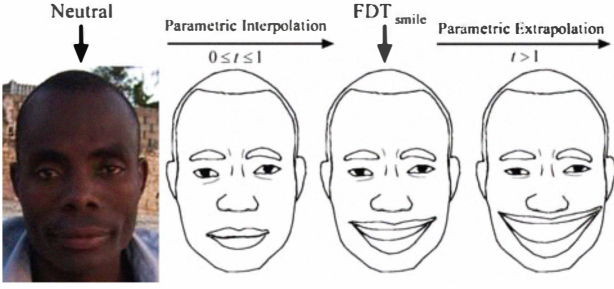


Fig. 5. Caricature exaggeration levels of the smile facial expression

## VI. CARICATURE RENDERING

To enhance the appearance of the generated caricature, we employ a stroke-based non-photorealistic rendering algorithm. The algorithm starts with a blank image (canvas), and then builds a composition of the caricature appearance, by progressively applying strokes along the rendering path, generated from section IV. Three main steps are involved in the preparation and composition of the caricature: (1) define the stroke locations, (2) compute stroke attributes, and (3) compose the strokes. The following sections explain each of the steps in more details.

### A. Stroke Locations Image

Every pixel along the rendering path, generated from section IV, serves as a reference to a stroke position along the path. To intensify the non-photorealistic effect of the final caricature composition, we randomly add extra stroke positions for every pixel along the rendering path (as shown in Fig. 6). The generated stroke locations are stored in a buffer as coordinates  $(x, y)$ . The stroke locations buffer is then used in later phases to compute and map the strokes into their position. An example image of the stroke locations buffer is shown in Fig. 7.

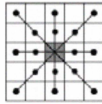


Fig. 6. Possible points to be randomly selected as a stroke location

### B. Stroke Attributes

In our approach, strokes are represented as rectangular shapes, as shown in Fig. 8. Each of the strokes has the following attributes: colour, position, orientation, and size. The colour of the strokes is based on the extracted rendering path, i.e., for the main facial features the stroke colour is black and for extra facial marks we use grey. The position of the stroke is acquired from the strokes locations buffer (section VI-A). We apply the geometric moment shape descriptors [13],

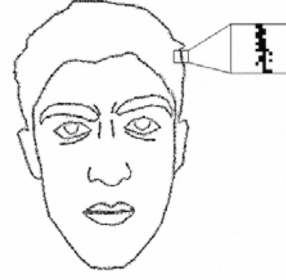


Fig. 7. Randomly generated stroke locations around the strokes rendering path.

along the rendering path, to determine the stroke's orientation  $\theta$  and height  $h$ , while the width  $w$  of the stroke is a value determined by the user (this is because the width of extracted path is very thin). In the context of the presented approach,

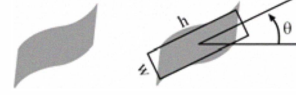


Fig. 8. A shape and its equivalent rectangle.

geometric moment shape descriptors are used to estimate the stroke feature parameters from local intensity distributions. The geometric moments for an  $(M \times N)$  image is given by [14]:

$$G_{pq} = \sum_x \sum_y x^p y^q I(x, y) \quad (7)$$

where  $M \times N$  are the width and the height of the image, and  $I(x, y)$  is the image intensity function.

The principal shape features (*width* ( $w$ ), *height* ( $h$ ), *centre coordinates* ( $x_c, y_c$ ), and *orientation*  $\theta$ ) are computed as follows:

$$x_c = \frac{G_{10}}{G_{00}}; y_c = \frac{G_{01}}{G_{00}} \quad (8)$$

$$\theta = \frac{\tan^{-1}(\frac{b}{a-c})}{2} \quad (9)$$

$$w = \sqrt{6(a+c-\Delta)}; h = \sqrt{6(a+c+\Delta)} \quad (10)$$

where  $a, b, c$  and  $\Delta$  are defined as,

$$\begin{aligned} a &= \frac{G_{20}}{G_{00}} - x_c^2; b = 2(\frac{G_{11}}{G_{00}} - x_c y_c) \\ c &= \frac{G_{02}}{G_{00}} - y_c^2; \Delta = \sqrt{b^2 + (a-c)^2} \end{aligned}$$

The above values can be used to map a brush stroke image onto the equivalent rectangle computed for an image segment. The stroke texture used can vary, for the results presented in this paper we used a thick colour pen as shown in Fig. 9 (d).



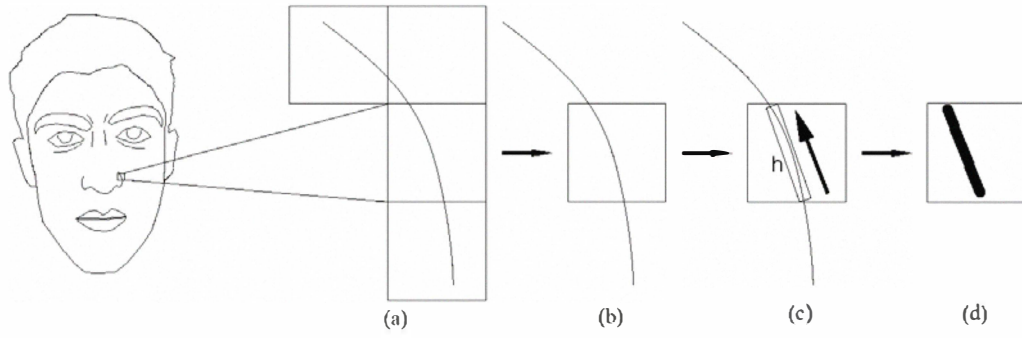


Fig. 9. The cropping process along the strokes rendering path. (a) shows a segment of the path subdivided into  $s \times s$  windows (b) cropping the window to be processed (c) apply geometric moments to find the stroke attributes (d) the initial stroke is painted along the path.

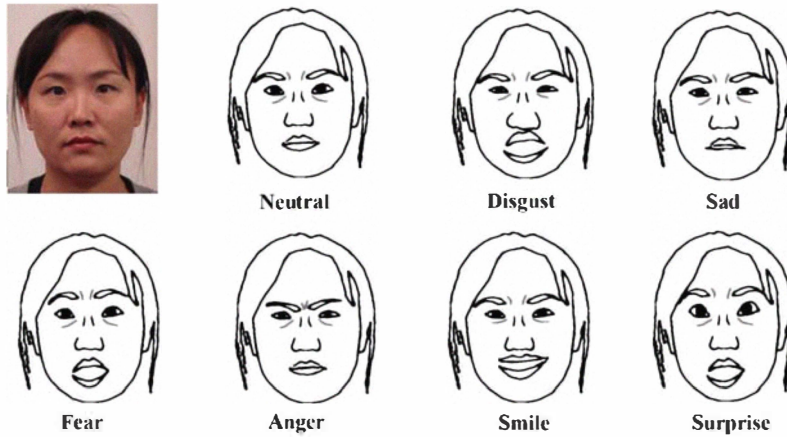


Fig. 10. Expressive caricatures generated from an input image.

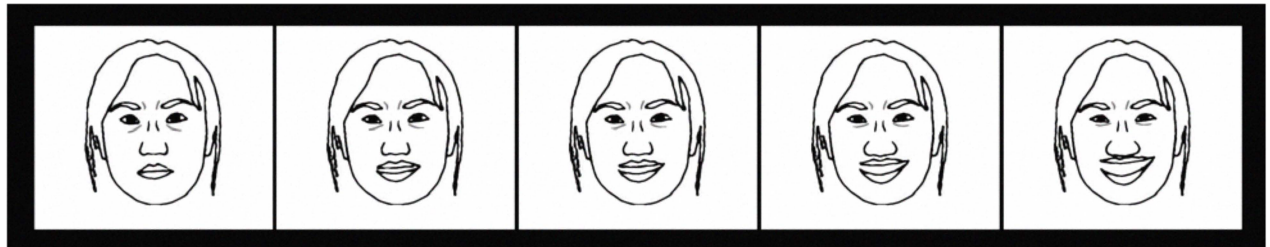


Fig. 11. A caricature animation sequence for the smile facial expression.

### C. Caricature composition

To render strokes along the drawing path, we use a coarse-to-fine rendering approach. Larger strokes are placed first along the strokes rendering path followed by smaller strokes around the rendering path. This ensures a complete rendering of the caricature with extra artistic effects. The following steps explain how the strokes attributes are computed:

**Large strokes :** We subdivide the extracted rendering path image into  $(s \times s)$  windows, where  $s$  is the size of the window. Each segment is cropped, and then used to compute the stroke attributes for that part of the drawing path. Fig. 9 demonstrates the process.

**Small strokes :** The image is further subdivided into  $(s/2 \times$

$s/2)$  window segments for every extra location stored in the stroke locations buffer. Coordinates of the stroke locations serve as the centre point of the subdivided window segments. Each segment is cropped and used to compute the stroke attributes using geometric moment functions. The centre position of the stroke is the same as the location coordinates stored in the stroke location buffer.

After determining the attributes of all subdivided window segments, the strokes are successively rendered one by one, using alpha blending, on a white canvas. Finally, the composition process is complete when the inner regions of the eye pupils are rendered using a flood-fill algorithm with a black colour.

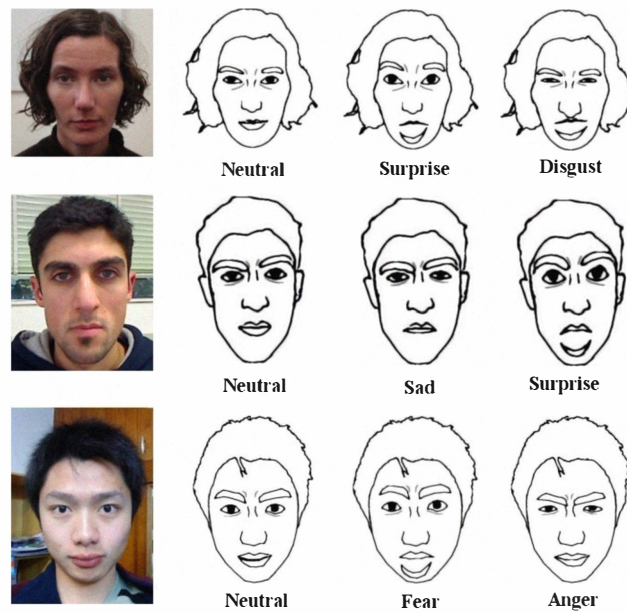


Fig. 12. Selected expressive caricatures generated by our system.

## VII. CARICATURE EXAMPLES

Fig. 10 illustrates an example of expressive caricatures generated using our proposed approach. The figure shows a neutral facial input image used to generate expressive caricatures of the facial expressions neutral, smile, surprise, disgust, anger, sad, and fear. The expressiveness of the caricature is controlled using the FDT's parameter values, while preserving the facial characteristics of the input image. Fig. 12 show more results of selected expressive caricatures from different individuals.

## VIII. CONCLUSION AND FUTURE WORK

We proposed a novel approach to generate artistically rendered expressive caricatures from an input neutral image. The approach consists of three main elements: (1) defining the rendering path using digital matting and AAM tracking, (2) expressivity control of the facial appearance using quadratic deformation models, and (3) stylized artistic rendering using a moment-based stroke rendering algorithm. The generated outputs show amusing expressive caricature results that are generated for the facial expressions neutral, smile, sad, fear, surprise, disgust, and anger. The achieved results are encouraging and can lead to future work in the field of expressive caricature generation.

Currently, our future directions for this research are directed towards conducting subjective evaluations of the generated caricatures to study their appearance, exaggeration, and the resembling of the facial expressions. Future work will also focus on further extending the describe approach to produce pencil sketch rendering of images and cartoon-like 3D characters.

## REFERENCES

- [1] A. Hughes and A. Gair, *Caricatures: Everything You Need to Know to Get Started*. London: HarperCollins Publishers, 1999.
- [2] S. Brennan, *Caricature Generator*. Cambridge, MIT: Master's thesis, 1982.
- [3] B. Gooch, E. Reinhard, and A. Gooch, "Human facial illustrations: Creation and psychophysical evaluation," *ACM Trans. Graph.*, vol. 23, no. 1, pp. 27–44, 2004.
- [4] H. Chen, N. Zheng, L. Liang, Y. Li, Y. Xu, and H. Shum, "Pictoon: a personalized image-based cartoon system," in *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*. New York, NY, USA: ACM, 2002, pp. 171–178.
- [5] P. Chiang, W. Liao, and T. Li, "Automatic caricature generation by analyzing facial features," *Asian Conference on Computer Vision*, 2004.
- [6] Y. Su, Y. Liu, U. Zhu, , and Z. Ren, "Facial sketch rendering and animation for fun communications," *Interactive technologies and sociotechnical systems: 12th International Conference, VSMM*, vol. 1, pp. 486–494, 2006.
- [7] M. Obaid, R. Mukundan, M. Billingham, and M. Sagar, "Facial expression representation using a quadratic deformation model," in *CGIV 2009: International Conference on Computer Graphics, Imaging and Visualization*. Tianjin, China: IEEE Computer Society, August 2009.
- [8] M. Sindeyev and V. Konushin, "A novel interactive image matting framework," *GraphiCon*, pp. 41–45, 2008.
- [9] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.
- [10] T. Kanade, Y. Yingli, and J. Cohn, "Comprehensive database for facial expression analysis," in *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*. Washington, DC, USA: IEEE Computer Society, 2000, p. 46.
- [11] T. Cootes, "Software, Modelling and Search Software," Retrieved From <http://personalpages.manchester.ac.uk/staff/timothy.f.cootes>, Dec. 2008.
- [12] J. Saragih and R. Goecke, "A nonlinear discriminative approach to AAM fitting," in *Proc. IEEE International Conference on Computer Vision ICCV2007*. Rio de Janeiro, Brazil: IEEE Computer Society, 2007, doi: 10.1109/ICCV.2007.4409106.
- [13] R. Mukundan and K. Ramakrishnan, "Moment functions in image analysis - theory and applications." Singapore: World Scientific Publishing Co. Pte Ltd, 1998.
- [14] M. Obaid, R. Mukundan, and T. Bell, "Enhancement of moment based painterly rendering using connected components," in *CGIV 2006: International Conference on Computer Graphics, Imaging and Visualization*. Sydney, Australia: IEEE Computer Society, 2006.